

Title : Database Table Version Unload  
Inventors : K. Pintar, J. Magill and K. Ziervogel  
Serial No. : 10/718,863 Filed : 21 November 2003  
Examiner : Khanh B. Pham Art Unit : 2166  
Docket : 149-0170US (03-022-US) Customer : 29855

Mail Stop Appeal Brief Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

## TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST .....	3
II.	RELATED APPEALS AND INTERFERENCES .....	3
III.	STATUS OF CLAIMS.....	3
IV.	STATUS OF AMENDMENTS.....	3
V.	SUMMARY OF CLAIMED SUBJECT MATTER .....	3
VI.	GROUND OF REJECTION TO BE REVIEWED ON APPEAL .....	4
VII.	ARGUMENT .....	4
A.	Section 102 Rejections (35 U.S.C. 102).....	5
1.	Hayashi (US 5,881,378) .....	5
2.	Comments on Hayashi .....	7
3.	Summary of Discussion Regarding Hayashi .....	8
B.	Section 103 Rejections (35 U.S.C. 103).....	8
C.	The Examiner's Allegation of Incomplete Response .....	8
D.	Conclusion.....	9
VIII.	CLAIMS APPENDIX .....	10
IX.	EVIDENCE APPENDIX .....	18
X.	RELATED PROCEEDINGS APPENDIX.....	19

**I. REAL PARTY IN INTEREST**

BMC Software, Inc. is the real party in interest.

**II. RELATED APPEALS AND INTERFERENCES**

None.

**III. STATUS OF CLAIMS**

Claims 1-26 and 30 stand rejected. Claims 27-29 have been withdrawn. Claims 1-26 and 30 are appealed.

**IV. STATUS OF AMENDMENTS**

None.

**V. SUMMARY OF CLAIMED SUBJECT MATTER**

This section provides a concise explanation of the subject matter defined in each independent claim involved in this appeal, referring to the specification by paragraph and line number and to the drawings by reference characters as required by 37 C.F.R. 41.37(c)(I)(v). *Note bene*, citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

Generally, Appellant claims a method (independent claim 1), program storage device (independent claim 14) and computer system (independent claim 30) to extract data from a database table, where the table includes at least two versions of data (*e.g.*, a current version and a prior version) and where each of the versions is associated with a different schema.

Specifically, independent claim 1 recites a database unload method comprising the acts of receiving a request to extract data from a database table, the database table having a current version associated with a current schema of the database table and a prior version associated with a prior schema of the database table, the request directed to the prior version (100, 105; ¶ 8, *ℓ.* 1-5; ¶ 12, *ℓ.* 1-2; ¶ 14, *ℓ.* 1-8; Table 1); and extracting data from the database table based on the table schema associated with the prior version (110, ¶ 16, *ℓ.* 1-3; 115, ¶ 17, *ℓ.* 1-2; 200, 205 ¶ 17, *ℓ.* 3-4; 210, ¶ 17, *ℓ.* 6-7; 215, ¶ 17, *ℓ.* 18-19; 220, ¶ 17, *ℓ.* 19-22; 120, ¶ 18, *ℓ.* 1-8).

Independent claim 14 is directed to a program storage device, readable by a programmable control device, comprising instructions stored thereon for causing the programmable control device to perform the method of independent claim 1 (¶ 21, *ℓ.* 5-16). Independent claim 30 is directed to a computer system for performing the method of independent claim 1.

## **VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 1-11, 14-24 and 30 stand rejected under 35 U.S.C. 102(b) as being anticipated by US Patent 5,881,378 to Hayashi et al. ("Hayashi"). Claims 12, 13, 25 and 26 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashi in view of US Patent 6,366,917 to St. John Herbert, III ("Herbert").

## **VII. ARGUMENT**

Independent claim 1 recites a method to retrieve a user-specified version of data from a database table that incorporates a plurality of versions/schemas. Independent claim 14 recites a program storage device containing instructions for performing the method of independent claim 1. Independent claim 30 recites a computer system for performing the method of independent claim 1. Accordingly, all claims stand or fall together. After a concise discussion of the Examiner's rejection and the cited art,

Appellant's arguments are presented below under separate headings as required by 37 C.F.R. 41.37(c)(I)(vii).

**A. Section 102 Rejections (35 U.S.C. 102)**

Claims 1-11, 14-24 and 30 stand rejected as being anticipated by Hayashi. With respect to independent claim 1, the Examiner alleges that:

Hayashi discloses:

A database unload method, comprising [database, extract logical information, Col 13 lines 5-10]:

receiving a request to extract data from a database table, the database table having a current version associated with a current schema of the database table and a prior version associated with a prior schema of the database table, the request directed to the prior version [request, old version, new version, database, Col 16 lines 55-60, Fig 17A-17B]; and

extracting data from the database table based on the table schema associated with the prior version [database, extract logical information, table, Col 13 lines 5-10, Fig 15A-B].

Final Office Action dated 30 Nov 2006 at page 5, ¶ 6. The Examiner has applied the same logic in rejecting independent claims 14 and 30. Final Office Action dated 30 Nov 2006 at pages 8 (¶ 6) and 9 (¶ 7).

**1. Hayashi (US 5,881,378)**

Hayashi is directed to a "derived database processing system." Hayashi at 1:14-19, see also Abstract.<sup>1</sup> As defined by Hayashi, a derived database is "a partial collection of components of [multiple] databases." Hayashi at 6:17-19 and Fig. 1 (element 18). The goal of a derived database is to be provide access to multiple databases as if they were a single database. Hayashi at 3:59-61, 10:14-16, 10 17-27 (Fig. 7) and 15:55-

---

<sup>1</sup> As used herein, the notation A:B-C denotes column A, lines B-C. Similarly, the notation A:B-C:D denotes column A, line B to column C, line D.

16:18. In other words, a “derived database” is another term for a “virtual database.” Hayashi at 6:59-61. Hayashi describes multiple embodiments that illustrate this interpretation. Hayashi at 7:25-63 and Fig. 2 (access to independently developed databases), 7:64-8:65 and Fig. 3 (access to a division database and a central database), 8:66-9:25 and Fig. 4 (access to databases having the same schema structure but operated differently), 9:26-53 and Fig. 5 (access to private and shared databases), 9:54-10:11 and Fig. 6 (access to a test database and a production database).

Hayashi describes the use of table schema, version or definition information only in the context of determining whether a first or “new” definition is consistent with a second or “old” definition and, if such consistency is found, to *replace* the old definition with the new definition. Hayashi at 16:53-17:9 and Fig. 11A. See also Hayashi at 17:30-37 (describing why a consistency check operation is useful) and 20:59-21:9 (describing a new definition operation in which only definition, not table data, is accessed and replaced). To this end, Hayashi teaches that an “access selecting unit 77 allows, *when definition information is being accessed*, either before-modification (old version) definition information or after-modification (new version) definition information to be selected. Hayashi at 18:63-6, Fig. 11A (emphasis added). To emphasize the fact that table data is not extracted from a database table based on a specified version of the table (as claimed), Hayashi explicitly states that “access selecting unit 77 *cannot be used* by an application program which simultaneously accesses to [sic] the definition information comprising both new and old version definition information, but can be used for verification of the new version definition information during the operation using the old version definition information.” Hayashi at 18:63-19-5, Fig. 11A (emphasis added). See also Hayashi at 19:40-20:8 and Figs. 12, 14A, 17A and 17B (describing accessing table definition information but declaring that “this does not allow a new version to co-operate with an old version” – that is, data access operations use only the most recent consistent version of the table schema to retrieve or extract data from a table).

## 2. Comments on Hayashi

The crux of the Examiner's rejection is based on his interpretation of a particular passage in Hayashi in which he alleges "Hayashi suggests extracting data form [sic] a table based on schema version."<sup>2</sup> Final Office Action dated 30 Nov 2006 at page 3, 3rd ¶; Office Action dated 14 June 2006 at page 4 (discussion of claim 1). The specific passage relied upon by the Examiner refers to Figure 9A and reads as follows:<sup>3</sup>

The logical information manipulating unit 53 outputs a CS identifier and an original name if it is an alias when a schema name and a table name is inputted to a derived database interpreting unit 14. The logical information manipulating unit 53 extracts logical information from table information. If there is a reference restriction defined in an SQL schema among tables, the logical structure information in the table(s) is also extracted. To update a table associated with the reference restriction, the reference restriction must remain unchanged. Hayashi at 13:4-13.

This passage says nothing about versions (*i.e.*, different schemas associated with a single table at different times). The phrase "logical information" refers to metadata used to map a logical name or identifier to a corresponding name or identifier in a physical table. Hayashi at 13:14-17. This distinction is made clear when Hayashi, in a subsequent paragraph, discusses the "storage information manipulating unit 54." Hayashi at 13:19-23. This unit manipulates the schema information of the underlying physical database tables. Thus, the logical information manipulating unit extracts *metadata* that defines the mapping between the name a user associates with a logical or virtual database (a "derived database" in the parlance of Hayashi) and a physical database table. The logical information manipulating unit does not unload data as recited by independent claims 1, 14 and 30. There is absolutely no discussion, or even

---

2 Applicant notes that a rejection based on 35 U.S.C. 102 *does not permit a mere suggestion*. "For a prior art reference to anticipate in terms of 35 U.S.C. 102, every element of the claimed invention must be identically shown in a single reference." *Diversitech Corp. v. Century Steps, Inc.*, 850 F.2d 675, 677, 7 U.S.P.Q.2d (BNA) 1315, 1317 (Fed. Cir. 1988). *See also* M.P.E.P. 2131.

3 The Examiner also relies upon Figures 15A-B (Office Action dated 14 June 2006 at page 4, last ¶) and 11A (Final Office Action dated 30 Nov 2006 at page 3, 3rd ¶). Neither of these figures support the Examiner's reasons for at least the same reasons as discussed herein.

hint, about table versions or of extracting data from a table based on the table's version – an aspect explicitly recited in the claimed invention. See independent claims 1, 14 and 30.

With respect to the term "reference restriction," one of ordinary skill in the art would understand that "reference restrictions" refer to referential integrity constraints defined within a physical data store or tables (*e.g.*, required relationships between primary and foreign keys in different tables). This passage simply says that if there are referential constraints associated with the physical tables that comprise a derived database, the derived database will behave so as to maintain these constraints. Again, there is absolutely no discussion here about table versions or of extracting data from a table based on the table's version.

### 3. Summary of Discussion Regarding Hayashi

As noted herein and in Applicant's prior Replies, Hayashi does not teach, describe or fairly suggest at least the claimed act of "extracting data from the database table based on the table schema associated with the prior version." For at least this reason, independent claims 1, 14 and 30 are patentable over Hayashi. For at least the same reason, dependent claims 2-11 and 15-24 are patentable over Hayashi as they depend from one of independent claims 1 and 14.

### **B. Section 103 Rejections (35 U.S.C. 103)**

Claims 12, 13, 25 and 26 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashi in view of US Patent 6,366,917 to St. John Herbert, III ("Herbert"). Final Office Action dated 30 Nov 2006 at page 5, ¶ 6. Each of claims 12, 13, 25 and 26 depend from one of independent claims 1 and 14. Accordingly, claims 12, 13, 25 and 26 are patentable over the cited prior art for at least the same reasons as independent claims 1 and 14.

### **C. The Examiner's Allegation of Incomplete Response**

The Examiner alleges "that Applicant's arguments do not comply with 37 C.F.R. 1.111(c) because they do not clearly point out the patentable novelty which ... the claims present in view of ... the references cited or the objections made. Further, they



do not clearly show how the claimed limitation avoids such references.” Final Office Action dated 30 Nov 2006 at page 3, 1st ¶.

Applicant strongly disagrees with the Examiner’s characterization. Applicant took great care to distinguish the claimed subject matter from Hayashi. Reply to Office Action filed on 7 Sep 2006 at pages 10-11, ¶ 2. Pointing out more than once that Hayashi does not teach at least the claimed act of “extracting data from the database table based on the table schema associated with the prior version.” Thus, Applicant has complied fully with 37 C.F.R. 1.111(c).

#### **D. Conclusion**

The Examiner has adopted an interpretation of Hayashi that is counter to the clear teaching of Hayashi. As shown above, Hayashi does not teach, describe or fairly suggest extracting *data* from a table based on a specified version of the table – that is, one of a plurality of schemas associated with the table. For at least this reason, the Examiner has failed to make a legitimate rejection under 35 U.S.C. 102 with respect to claims 1-11, 14-24 and 30. Further, the Examiner’s rejection of claims 12, 13, 25 and 26 under 35 U.S.C. 103 is moot in so far as each of these claims depend from one of independent claims 1 and 14. Accordingly, Applicant respectfully requests the Panel reverse the Examiner’s rejections and permit claims 1-26 and 30 to issue.

Respectfully submitted,

/Coe F. Miles, Ph.D., J.D./

Reg. No. 38,559

**Wong, Cabello, Lutsch, Rutherford & Brucculeri, L.L.P.**

Customer No. 29855	Voice: 832-446-2418
20333 SH 249, Suite 600	Mobile: 713-502-5382
Houston, Texas 77070	Facsimile: 832-446-2458
Email: cmiles@counselIP.com	

## **VIII. CLAIMS APPENDIX**

1. A database unload method, comprising:  
  
receiving a request to extract data from a database table, the database table having a current version associated with a current schema of the database table and a prior version associated with a prior schema of the database table, the request directed to the prior version; and  
  
extracting data from the database table based on the table schema associated with the prior version.
2. The method of claim 1, wherein the act of receiving a request further comprises obtaining schema definition information associated with the database table.
3. The method of claim 2, wherein the act of obtaining schema definition information comprises obtaining schema definition information for the prior version.
4. The method of claim 3, wherein the act of obtaining schema definition information further comprises obtaining schema definition information for versions associated with the database table in addition to the prior version.

5. The method of claim 2, wherein the act of obtaining schema definition information comprises receiving said schema definition information from a user.
6. The method of claim 2, wherein the act of obtaining schema definition information comprises receiving said schema definition from a database change management application.
7. The method of claim 2, wherein the act of obtaining schema definition information comprises receiving said schema definition information directly from a database management system.
8. The method of claim 1, wherein the act of extracting data comprises unloading data stored in the database table to a result set data structure.
9. The method of claim 8, wherein the result set data structure comprises a computer file.
10. The method of claim 1, wherein the act of extracting data comprises generating a file that encodes therein a definition of the schema associated with the prior version.

11. The method of claim 1, wherein the act of extracting data comprises:  
unloading a datum from the database table, said datum having a first format;  
and  
transforming the unload datum to a second format.
12. The method of claim 1, wherein the act of extracting data comprises:  
identifying a row in the database table;  
determining a version associated with the identified row; and  
extracting data from the identified row in accordance with the determined  
version.
13. The method of claim 12, wherein the acts of identifying, determining and  
extracting are repeated for each row in the database table.

14. A program storage device, readable by a programmable control device, comprising instructions stored on the program storage device for causing the programmable control device to:

receive a request to extract data from a database table, the database table having a current version associated with a current schema of the database table and a prior version associated with a prior schema of the database table, the request directed to the prior version; and

extract data from the database table based on the table schema associated with the prior version.

15. The program storage device method of claim 14, wherein the instructions to receive a request further comprise instructions to obtain schema definition information associated with the database table.

16. The program storage device of claim 15, wherein the instructions to obtain schema definition information comprise instructions to obtain schema definition information for the prior version.

17. The program storage device of claim 16, wherein the instructions to obtain schema definition information further comprise instructions to obtain schema definition information for versions associated with the database table in addition to the prior version.

18. The program storage device of claim 15, wherein the instructions to obtain schema definition information comprise instructions to receive said schema definition information from a user.

19. The program storage device of claim 15, wherein the instructions to obtain schema definition information comprise instructions to receive said schema definition from a database change management application.

20. The program storage device of claim 15, wherein instructions to obtain schema definition information comprise instructions to receive said schema definition information directly from a database management system.

21. The program storage device of claim 14, wherein the instructions to extract data comprise instructions to unload data stored in the database table to a result set data structure.

22. The program storage device of claim 21, wherein the instructions to unload data to a result set data structure comprise instructions to unload data to a computer file.

23. The program storage device of claim 14, wherein the instructions to extract data comprise instructions to generate a file that encodes therein a definition of the schema associated with the prior version.

24. The program storage device of claim 14, wherein the instructions to extract data comprise instructions to:

unload a datum from the database table, said datum having a first format; and  
transform the unload datum to a second format.

25. The program storage device of claim 14, wherein the instructions to extract data comprise instructions to:

identify a row in the database table;  
determine a version associated with the identified row; and  
extract data from the identified row in accordance with the determined version.

26. The program storage device of claim 25, wherein the instructions to identify, determine and extract are repeated for each row in the database table.

27. A relational database data-unload command, comprising:
- a table-identifier to identify a table in a relation database from which to unload data;
  - a version-identifier to identify a version of the table identified by the table-identifier the data-unload command is to be executed against.
28. The relational database data-unload command of claim 27, further comprising one or more column-identifiers to specify the columns to unload from the table identified by the table-identifier.
29. The relational database data-unload command of claim 28, further comprising one or more transformation-identifiers to specify a function to apply to a datum unloaded from a specified column of the table identified by the table-identifier.



30. A computer system, comprising:

a central processing unit;

first storage operatively coupled to the central processing unit, the first storage having stored therein at least a portion of a relational database table; and

second storage operatively coupled to the central processing unit and the first storage, the second storage having stored therein at least a portion of a database management system, the database management system adapted to -

receive a request to extract data from the relational database table, the relational database table having a current version associated with a current schema of the relational database table and a prior version associated with a prior schema of the relational database table, the request directed to the prior version, and

extract data from the relational database table based on the table schema associated with the prior version.

**IX. EVIDENCE APPENDIX**

None.

**X. RELATED PROCEEDINGS APPENDIX**

None.